

Tecnologias e Programação de Sistemas de Informação

Vetores multidimensionais e estratégias de resolução de problemas

Arquitetura de Dispositivos | David Jardim

Cofinanciado por:



Da aula anterior...

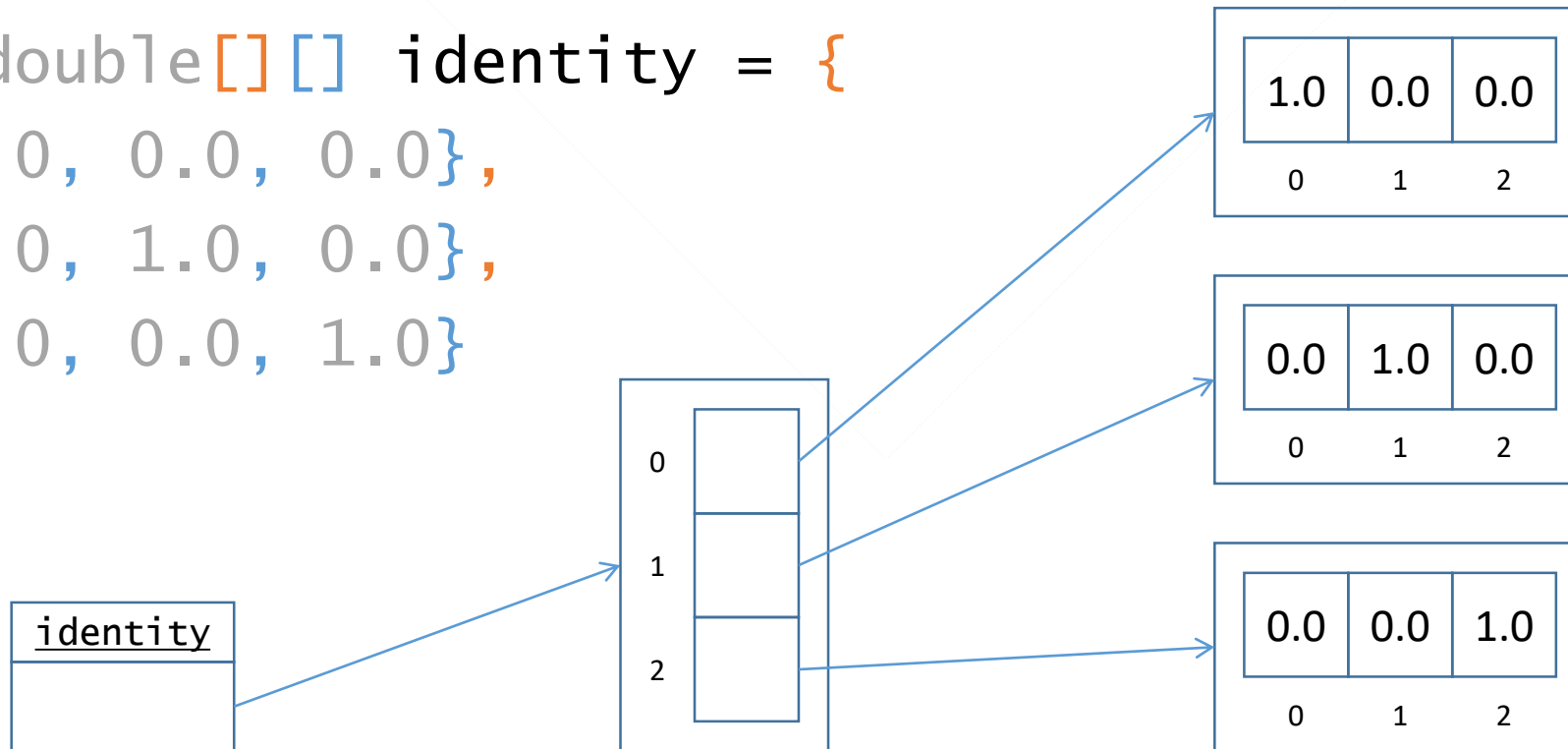
- Manipulação de caracteres
- Relação entre os tipos `char` e `int`
- Tipos de valor vs. tipos de referência

Vetores multidimensionais

- Itens de vetor podem ser vectores
- Possíveis vetores de mais do que uma dimensão
- Dimensões
 - 1D: `int[] grades;`
 - 2D: `int[][] gradesPerCourse;`
 - 3D: `int[][][] gradesPerCoursePerProgram;`
 - Etc.

Exemplo regular: matriz identidade de 3×3

```
final double[][] identity = {
    {1.0, 0.0, 0.0},
    {0.0, 1.0, 0.0},
    {0.0, 0.0, 1.0}
};
```



Exemplo irregular: factores primos de primeiros naturais

```
final int[][] primeFactors = {
    null,
    {},
    {2},
    {3},
    {2, 2},
    {5},
    {2, 3},
    {7},
    {2, 2, 2},
    {3, 3},
    {2, 5}
};
```

O vector `primeFactors` contém vectores com diferentes tamanhos. Para obter os factores primos de um dado natural $1 \leq n \leq 10$ usa-se `primeFactors[n]`, que é um simples vector de inteiros. Como 0 (zero) não tem factores primos, o primeiro vector não existe. Isso representa-se usando uma referência nula.

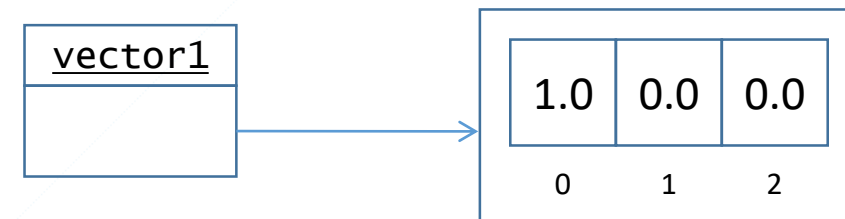
Os factores primos de um inteiro positivo são os números primos que dividem esse inteiro exactamente

180	2
90	2
45	3
15	3
5	5
1	

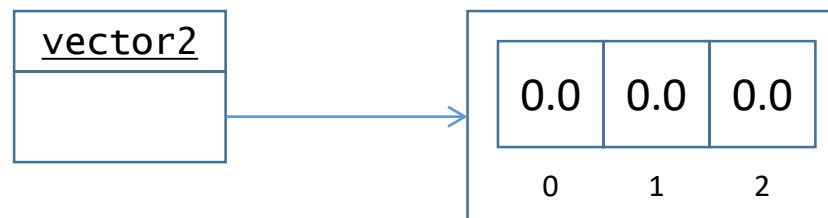
$$2^2 \times 3^2 \times 5 = 180$$

Construção 1D

```
final double[] vector1 = {1.0, 0.0, 0.0};  
final double[] vector1 =  
    new double[] {1.0, 0.0, 0.0};
```

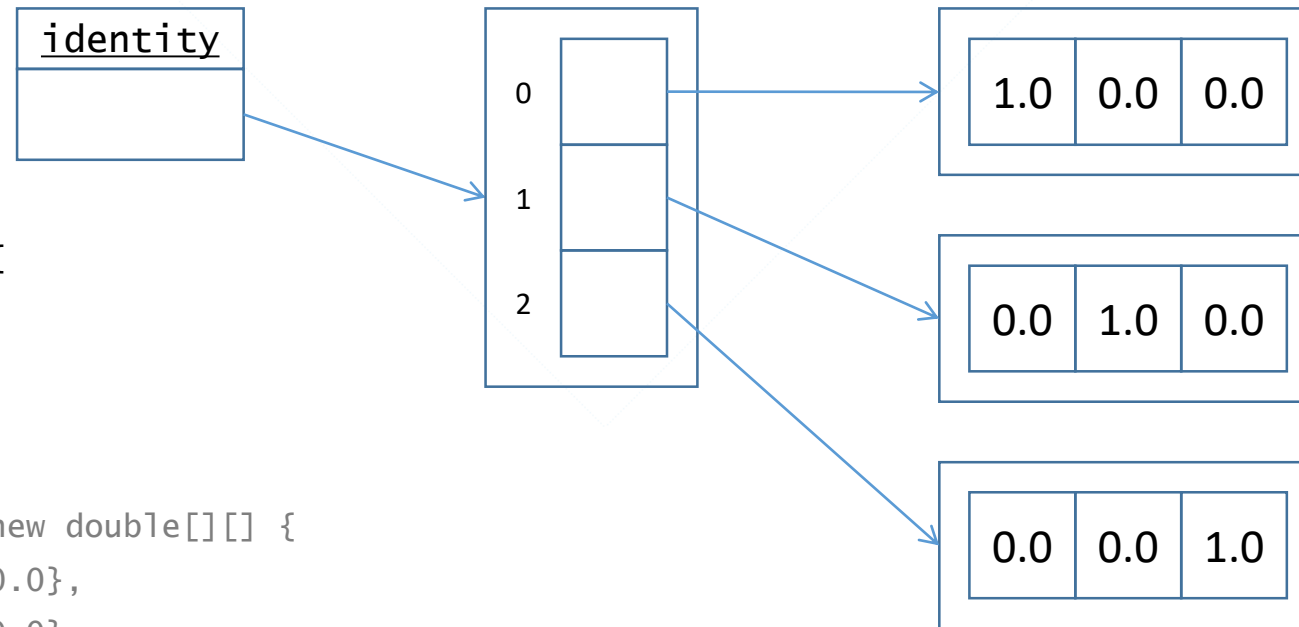


```
final double[] vector2 = new double[3];
```



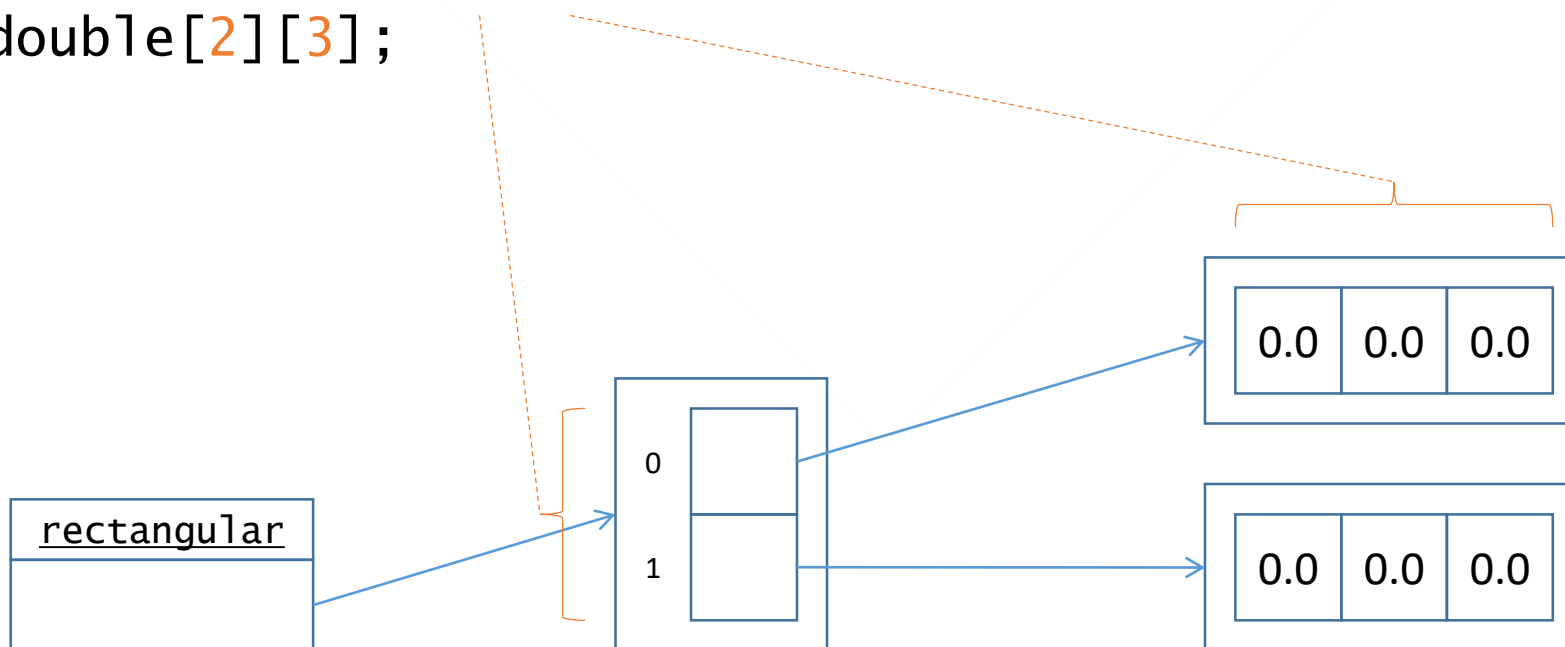
Construção 2D: regular

```
final double[][] identity = {  
    {1.0, 0.0, 0.0},  
    {0.0, 1.0, 0.0},  
    {0.0, 0.0, 1.0}  
};  
  
final double[][] identity = new double[][] {  
    new double[] {1.0, 0.0, 0.0},  
    new double[] {0.0, 1.0, 0.0},  
    new double[] {0.0, 0.0, 1.0}  
};
```



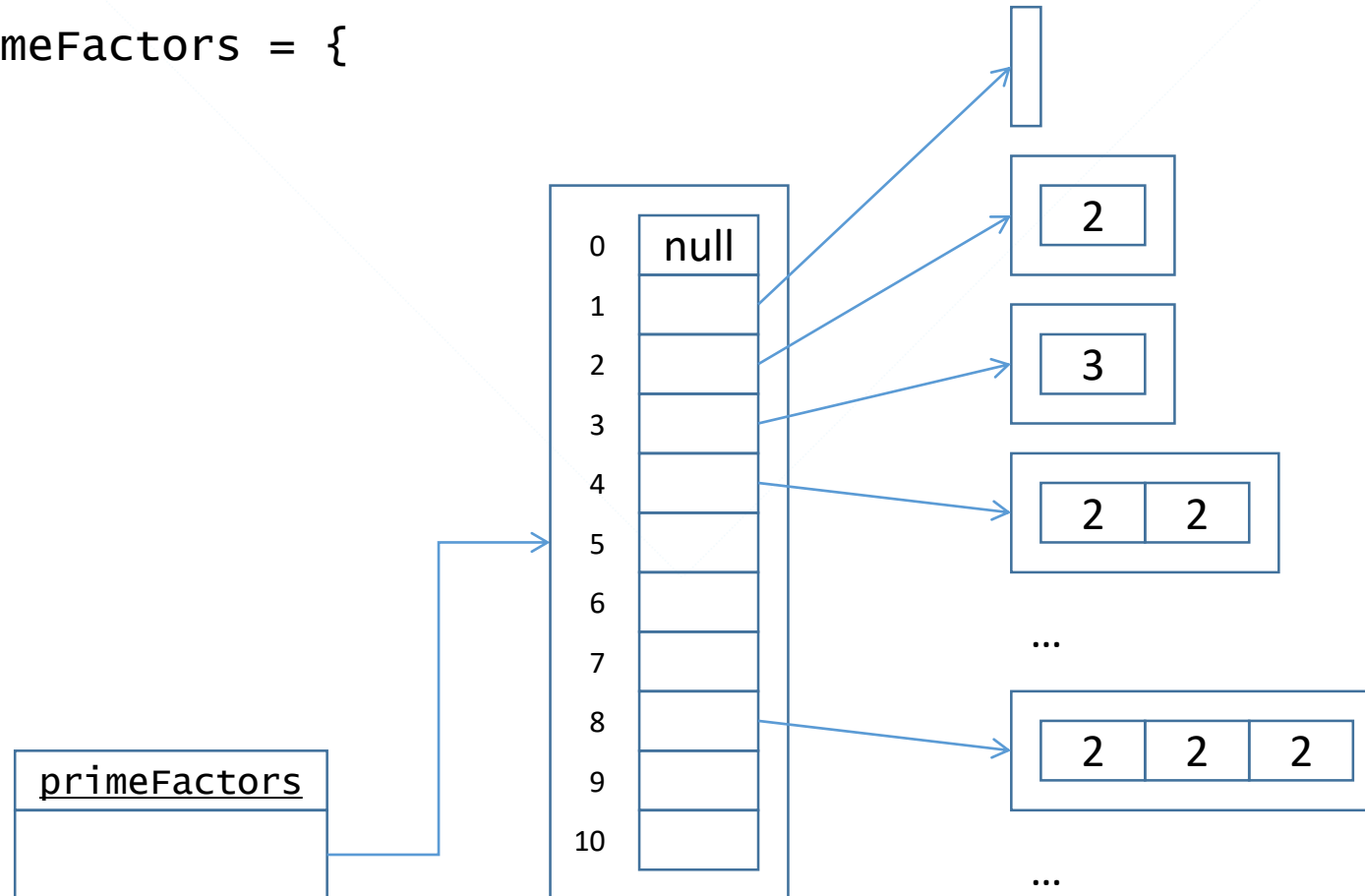
Construção 2D: regular

```
final double[][] rectangular =  
    new double[2][3];
```

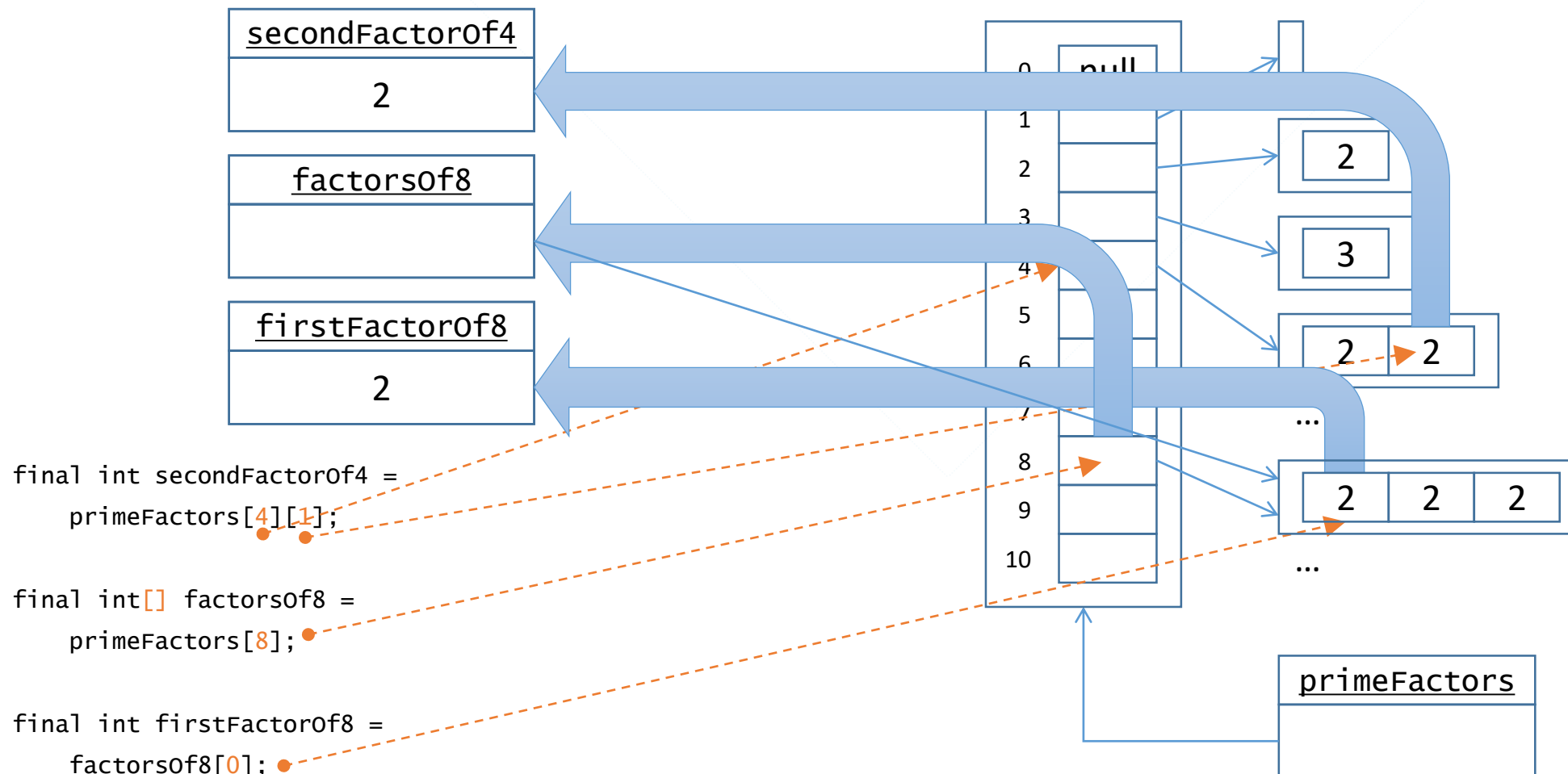


Construção 2D: irregular

```
final int[][] primeFactors = {
    null,
    {},
    {2},
    {3},
    {2, 2},
    {5},
    {2, 3},
    {7},
    {2, 2, 2},
    {3, 3},
    {2, 5}
};
```



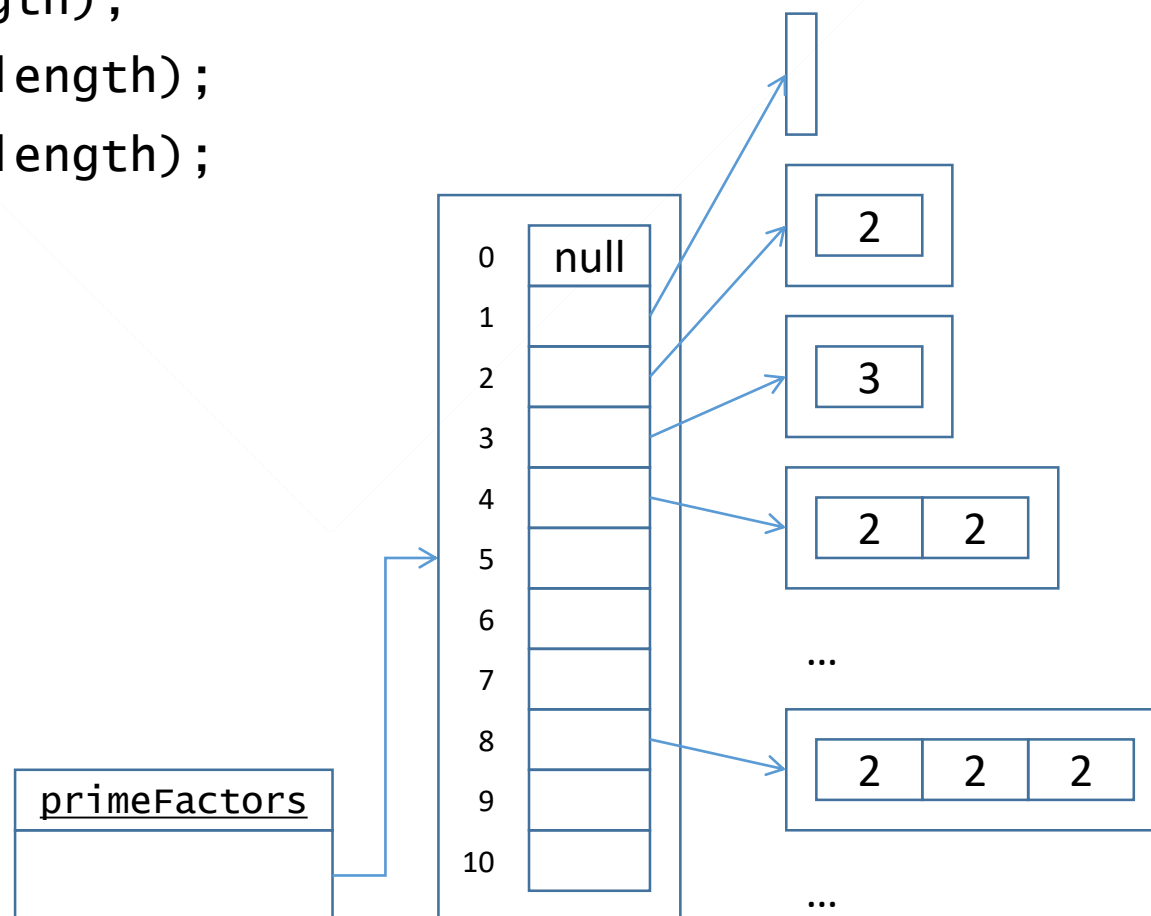
Indexação



Tamanhos

```
out.println(primeFactors.length);
out.println(primeFactors[2].length);
out.println(primeFactors[8].length);
```

```
11
1
3
_
```



Exemplo: médias

```
int[][] gradesPerCourse = {  
    {14, 16},  
    {12, 18, 20}  
};  
  
int numberOfGrades = 0;  
double sumOfGrades = 0.0;  
for (int course = 0; course != gradesPerCourse.length; course++){  
    for (int i = 0; i != gradesPerCourse[course].length; i++) {  
        sumOfGrades += gradesPerCourse[course][i];  
        numberOfGrades++;  
    }  
}  
double averageOfGrades = sumOfGrades / numberOfGrades;
```

Problema

Dados

1. um vector com números de alunas(os),
2. uma matriz com as siglas das UC e
3. uma matriz de notas (em que cada linha corresponde às notas que a(o) aluna(o) correspondente tem nas UC indicadas, uma por coluna),

escrever código Java que

1. mostre as notas de cada aluna(o),
2. mostre a média das notas de cada aluna(o) e
3. mostre a média mais alta, identificando a quem pertence.

```
final int[] studentNumbers = {20032, 30312, 25342, 31432};  
final String[] courseAcronyms = { "AD", "WEB-FE", "WEB-BE"};  
final int[][] studentGrades = {  
    {15, 18, 17},  
    {18, 10, 11}, // grades of student 30312  
    {11, 13, 15},  
    {10, 19, 16}  
};
```

Resolução de problemas: etapas

Etapa	Descrição/notas
Compreender o problema!	Ser cuidadoso e preciso.
Planear	<ul style="list-style-type: none">• (Escolher uma ou mais estratégias.)• Partir a tarefa em pequenos passos.• Descrever em pormenor cada passo.
Seguir o plano	<ul style="list-style-type: none">• Decidir qual o próximo passo.• Executar o próximo passo.
Rever e estender	<ul style="list-style-type: none">• Analisar o processo de resolução.• Aprender com os erros.

Resolução de problemas: estratégias

Estratégia	Descrição
Analogia	Conhecemos soluções para problemas semelhantes?
Generalização	O nosso problema é caso particular de outro de que conhecemos a solução?
Especialização	O nosso problema é um caso genérico de outro mais particular de que conhecemos a solução?
Indução	Compreendemos melhor o problema se estudarmos alguns dos seus casos particulares?
Alteração	Será que alterar o enunciado do problema o transforma noutro de que conhecemos a solução?

Resolução de problemas: estratégias

Estratégia	Descrição
Aproximações sucessivas	<ul style="list-style-type: none">• Será que conseguimos encontrar uma forma de atingir um subobjectivo e ficar, assim, mais próximo da solução?• Será que podemos repetir o processo até resolver o problema?
Decomposição ou dividir para conquistar	Será que podemos dividir o problema em subproblemas de mais simples solução e compor, assim, uma solução global?
Partir do objectivo	Será que olhar para o nosso destino nos dá boas pistas acerca do caminho a seguir?
Fazer um diagrama/desenho	Nem calcula o quanto ajuda...

Mais informação

- Resolução de problemas
 - [http://en.wikipedia.org/wiki/Problem solving](http://en.wikipedia.org/wiki/Problem_solving)
 - [http://en.wikipedia.org/wiki/How to Solve It](http://en.wikipedia.org/wiki/How_to_Solve_It)

A reter

- Vectores multidimensionais
- Estratégias de resolução de problemas

